



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F	A2	(11) International Publication Number: WO 98/47057 (43) International Publication Date: 22 October 1998 (22.10.98)
(21) International Application Number: PCT/US98/07495 (22) International Filing Date: 14 April 1998 (14.04.98) (30) Priority Data: 60/043,643 14 April 1997 (14.04.97) US (71) Applicant (for all designated States except US): R2K, LLC [US/US]; 53 East 34th Street, New York, NY 10016 (US). (72) Inventors; and (75) Inventors/Applicants (for US only): YIEN, Richard [US/US]; 343 East 30th Street, New York, NY 10016 (US). RUBINO, Joseph [US/US]; 512 Pat Drive, West Islip, NY 11795 (US). SABIN, Todd [US/US]; 16 P Reading Road, Edison, NJ 08817 (US). DUDA, Jacek [US/US]; 93-05 68th Avenue, Forest Hills, NY 11375 (US). POLIVKA, Andrej [US/US]; Apartment H-A-1, 208 Harris Road, Bedford Hills, NY 10507 (US). (74) Agents: GARNER, Melvin, C. et al.; Darby & Darby P.C., 805 Third Avenue, New York, NY 10022-7513 (US).		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published <i>Without international search report and to be republished upon receipt of that report.</i>
(54) Title: REGISTRY MANAGEMENT SYSTEM (57) Abstract <p>A configuration management system and method manage configuration settings and representations thereof for computers in a network, with computers grouped according to common setting values. The configuration settings are stored in a configuration database, which is compared to the configuration settings of each computer during operation of the network. Computer configurations as Already-Have lists are scanned and common settings are extracted to generate Should-Have lists, which may in turn be scanned for common settings among groups to generate sublists and sub-sublists. As discrepancies and conflicts in settings are detected, the configuration setting system and method compares Should-Have lists, determines differences, and displays such differences to a user such as an administrator. The conflicting differences may be automatically resolved by either automatically changing the state of the representation in the configuration database or by changing the current configuration settings of particular computers associated with the different settings. Alternatively, manual intervention by users may be initiated to override automatic conflict resolution. Should-Have list precedence may also be used to control configuration setting conflicts, in which a Should-Have list more closely associated with an individual computer takes precedence of an encompassing Should-Have list of a group of computers or of the entire network, so that individual computers may have customized configurations, with the network configuration controller recognizing such customized configurations yet maintaining network operability.</p>		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon	KR	Republic of Korea	PL	Poland		
CN	China	KZ	Kazakhstan	PT	Portugal		
CU	Cuba	LC	Saint Lucia	RO	Romania		
CZ	Czech Republic	LI	Liechtenstein	RU	Russian Federation		
DE	Germany	LK	Sri Lanka	SD	Sudan		
DK	Denmark	LR	Liberia	SE	Sweden		
EE	Estonia			SG	Singapore		

5

10

REGISTRY MANAGEMENT SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This patent application claims the priority of U.S. provisional patent application No. 60/043,643, which is incorporated herein by reference.

15

BACKGROUND OF THE INVENTION

a) Field of the Invention

This invention generally relates to management systems for controlling configuration settings to each computer throughout a network, and more particularly, to a management system for conforming the configuration files in a network of computers using the "WINDOWS 95" or "WINDOWS NT" operating systems.

20

b) Description of the Related Art

As personal computers have become increasingly connected via mechanisms such as local area networks, intranets and the Internet, more and more application software has been developed which leverages such connectivity. Examples of such application software range from the basic functionality of shared file and printer services to groupware calendar scheduling applications to complex client/server database applications. As networked applications have grown to be more popular and more complex, their system administration requirements have also increased. Shared applications require that someone centrally determine and administer for a

25

30

group of personal computers and users, functions such as access, security, shared data and resources, performance optimization, and common application attributes. In many situations, it is critical for groups of machines to have complementary or parallel configuration settings, in the absence of which the application software becomes dysfunctional or performance suffers at a local site or throughout the network. Diagnosing an application in which the configuration settings are faulty can require many hours of a highly-skilled person's time to identify the root cause of the problem from many possible causes. Often, the administrator must physically access the individual personal computers to examine and/or change faulty configuration settings. This can be a logistical problem when the computers in question are distributed over many geographical locations.

U.S. Patent No. 5,414,812, issued to Filip et al., discloses an object-oriented, hierarchical model of a computer network layered communications subsystem implemented in a configuration database subsystem to create and maintain a configuration database and to provide configuration data to the layered communications subsystem. The layered communication subsystem is represented by a set of defined object classes. Each object class corresponds to one or more functions defined at each of the communications subsystem layers. Each of the object classes is related in a hierarchical relationship which preserves the functional relationship among the various functions in the various layers making up the layered communications subsystem. The configuration database subsystem provides a user interface for command and data input, a configuration support module which implements command processing, validation and object support programs which execute the input commands and allow a configuration database to be built and maintained while ensuring the integrity of the configuration data and the hierarchical relationship, and a database for storing the configuration.

U.S. Patent No. 5,588,147, issued to Neeman et al., discloses a method for providing a plurality of files organized into a tree of files in a distributed system having a

replication facility and a number of computer systems. A file or a subtree of files at one of the computers can be replicated and stored in another computer.

U.S. Patent No. 5,432,941, issued to Crick et al., discloses a dynamic configuration software system wherein a multiplicity of software routines are defined. During configuration of the software system, a plurality of the software routines are selected for inclusion in the software system. Before configuring the software system, each of the multiplicity of software routines is assigned a configuration group. When configuring the software system, each of the multiplicity of software routines is invoked, one configuration group at a time. Each invoked software routine determines whether it should be included in the software system, and if it is to be include, the software routine pushes a reference to itself on a call-down table. Thus, by changing the assigned configuration order of a software routine, the configuration of the software system can be changed.

U.S. Patent No. 5,581,764, issued to Fitzgerald et al., discloses an enterprise management system for a plurality of desktop computers. The system "reads" each computer and extracts an Already-Have List of configuration data. The system then compares the Already-Have List of each computer with a Should-Have List for that computer. The system detects differences between the two lists and automatically downloads to each particular computer any data located in the Should-Have List which is not present in the Already-Have List. The system commands the changes to the Already-Have List of each computer and does not follow rules of precedence and inheritance.

U.S. Patent No. 5,247,683, issued to Holmes et al., discloses a system for installing software and updating configuration files. The system reads configuration build files of new software, for example (which are essentially "Should-Have Lists") and also reads configuration build files of the installed software (which are essentially "Already-Have Lists"). If there are any "clashing statements" between the two configuration build files, the system according to U.S. Patent No. 5,247.683 will compare "priority values" assigned to each clashing

statement and automatically discard the lower priority statement. After all clashing statements are resolved, the system replaces the configuration files of the machine.

5 A problem with the system disclosed in U.S. Patent No. 5,247,683 is that conflicts are not resolved at the local level, they are determined by a preassigned priority value which relates to the type of configuration setting in conflict. The system does not recognize the specific configuration requirements of each individual computer.

10 Other disadvantages of prior art systems in managing configuration files is that such systems tend to be inefficient and fail to recognize that the goal of any of these systems is to ensure that each computer within the network operates as it should, and that all of the computers should operate together as expected or desired.

OBJECTS OF THE INVENTION

15 It is therefore an object of the present invention to provide a system for managing the configuration files of each of a plurality of computers throughout an enterprise, which system overcomes the deficiencies of prior art systems.

It is another object of the invention to provide such a configuration management system which efficiently groups computers together according to specific requirements as determined at the local level.

20 It is another object of the invention to provide such a configuration management system which resolves conflicting configuration settings using human intervention at the local level.

It is yet another object of the invention to provide such a configuration management system which follows the rules of inheritance to effectively group related computers together, and precedence to resolve conflicting management values.

SUMMARY OF THE INVENTION

5 The present invention is a software system with facilities to create, store, organize, and revise configuration settings from a central location and then to distribute and reconcile such settings to multiple "local" networked computers. The system uses a logical database of computer configuration settings for which common system settings for computers in a related subtree of the groups are determined, versioned, and stored. The configuration settings can be extracted from individual computer configurations, and configuration integrity is maintained and managed by determining differences in configuration settings and resolving such differences by accessing the computers directly using remote network access functions or by transmitting to a change distribution system, such as the "MICROSOFT" "SYSTEMS MANAGEMENT SERVER", for implementation at a scheduled time. In addition, the calculated configuration settings can be compared to the existing configuration settings for a particular computer with differences displayed to a user for diagnostic purposes and to allow the user to accept or refuse a particular configuration change.

15 The present system allows a configuration manager in communication with a user at a local-level site to question a particular configuration setting originating from a group level site (or an enterprise level site) and provides an opportunity to override the configuration setting. The configuration manager may also provide an explanation for the override to the group level site (or enterprise level site). For example, one particular setting may work better in the specific local computing environment as learned through personal knowledge or experience obtained at the local level (or machine level) than the group level configuration. Since the reason for the local selection is sent to the group or enterprise level, valuable experience regarding operations at the local level may be put to use throughout the entire network.

25 According to the invention, if a new configuration setting conflicts with an existing setting located at a first level, generally, the existing configuration setting, i.e., the one "closer" (as defined below) to the local site, takes precedence over the new configuration setting.

According to another aspect of the invention, an "allow-override" value for each (or selected) configuration settings is provided to dictate which configuration settings are allowed to be overridden at the local level.

5 In an exemplary embodiment of the invention the software system comprises a database including a representation, such as a Registry, of groups of computers and their desired configuration settings, a program that enables an administrator to update this representation, and a program which will examine the current computer configuration settings for each computer in the database and compare the current settings with new configuration settings that are earmarked for that computer in order to produce a list of differences.

10 In addition, the program can automatically resolve any conflict between the representation or Registry entries of the computer in a main set of configuration settings and the current configuration settings of the particular computer by either changing the state of the representation or by changing the current configuration settings of the computer.

15 BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other features of the present invention will be more readily apparent from the following detailed description and drawings of an illustrative embodiment of the invention in which:

20 Fig. 1 is a block diagram of an enterprise network system showing a configuration server and a plurality of computers, interconnected by a bus, according to the invention;

Fig. 2 is a block diagram of four representative computers showing details of selected configuration settings, utilizing an illustrative embodiment of the present invention;

Fig. 3 is a block diagram showing computers and groups thereof according to the present invention;

25 Fig. 4 is a block diagram showing computers and groups arranged in a hierarchy, according to the present invention;

Fig. 5 is a flow chart of the method of operation of the software system of an illustrative embodiment of the invention;

Fig. 6 is a flow chart of the method of operation for determining differences in configuration settings;

5 Fig. 7 is a portion of a flow chart showing the method of operation for determining differences in configuration settings and for determining an "allow override" setting; and

Fig. 8 is a block diagram showing computers and groups of computers arranged in a hierarchy to illustrate the rules of precedence, according to the invention.

DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

Referring to Fig. 1, the present invention is directed to configuration management system 10 and method of use thereof which function in network 12. Network 12 includes a plurality of machines 14, i.e., computers such as personal computers (PCs), servers, private
15 branch exchanges (PBXs) and other computing devices, which may be labelled A-D, respectively. Without loss of generality, machines 14 are referred to herein as computers, which are connected over network connections 16 to configuration server 18, and optionally to each other, such as in a token ring network configuration. Configuration server 18 includes or is connected to memory such as configuration database 20. In an illustrative embodiment, network
20 12 operates with computers 14 and manages data transfers and transactions by running the "MICROSOFT" "WINDOWS NT" operating system. Network connections 16 may be a local area network (LAN) for interconnecting computers 14 to configuration server 18 and optionally to each other.

25 Configuration management system 10 of the present invention may be implemented in software executed by configuration server 18, or alternatively may be hardware components of or integrated features of configuration server 18 and/or components operatively

connected thereto in order to allow an administrator to modify the configuration settings of network 12 and of the plurality of computers 14. Configuration management system 10 of the present invention may be an application program in which case it may be compiled or assembled from source code written, for example, in C++ or "VISUAL BASIC". Alternatively, configuration management system 10 may be placed in the network operating system (NOS). A computer or terminal of an administrator may include a display, and configuration management system 10 of the present invention may also generate or operate in conjunction with a graphic user interface (GUI) such as the GUI of "WINDOWS NT", to allow an administrator to point-and-click or otherwise provide commands and selections to reconfigure network 12 and computers 14 thereof.

Configuration settings for network 12 are typically stored in a memory such as configuration database 20, with configuration server 18 capable of allowing an administrator to modify the configuration settings through an administrative access point, which may be a terminal and/or may be one of computers 14. Such configuration settings may be stored in configuration database 20 in at least one computer file. In an illustrative embodiment, the present invention is described in terms of the "MICROSOFT" "WINDOWS NT" network operating system (NOS). Accordingly, configuration management system 10 of the present invention is capable of modifying the configuration settings in configuration database 20. Network 12, in an illustrative embodiment, uses a configuration file which is common to the configuration of each computer 14, with the configuration file being called the Registry in the "MICROSOFT" "WINDOWS NT" network operating system. The configuration file is a central repository for configuration settings used by virtually all software applications written for "WINDOWS NT". The "WINDOWS NT" operating system contains built-in functions which allow a program to manipulate the entries in the Registry by adding, modifying, and deleting entries, the functions being called "Akeys" and "Anames", and their associated values, called "data". Each computer 14, in turn, includes a Registry associated with the attributes specific

to the particular computer 14. Accordingly, configuration management system 10 of the present invention is capable of modifying the Registry of the entire network 12 and of each of computers 14.

It is understood that other networks and network operating systems may be used, and so the present invention is not limited to the "MICROSOFT" "WINDOWS NT" NOS and thus is not limited to reconfiguring network 12 and computers 14 thereof by reconfiguring a Registry. Configuration files appropriate to a non-"MICROSOFT"-based NOS may be modified by an administrator using configuration management system 10 and the method of use described herein.

For illustrative purposes herein, configuration server 18 manages the Registry settings of each "WINDOWS NT" computer 14 throughout network 12. Configuration database 20 stores relationship information regarding each computer 14 on network 12, groups of computers, and linking information, as described below. Configuration server 18 uses configuration management system 10, embodied as an executable program, according to the invention, and uses configuration database 20 to perform configuration management functions.

Configuration management system 10 according to the invention provides for the efficient management of system configuration settings (e.g., in the Registry file in "WINDOWS NT" and "WINDOWS 95") for computers 14 within network 12 and includes facilities to create, store, organize, and revise such settings from configuration server 18. According to the disclosed invention, information regarding each computer 14 is collected and stored in configuration database 20. Such information includes computer identity, group names, current Registry settings within each computer 14, and linking information which links or associates any group or computer with other computers in a parent/child relationship. The following tables illustrate the above-described information:

A. Database Tables

1. Computers and Groups

The Computers and Groups Table has the following attribute definition:

NAME	TYPE	DEFINITION
Type	MACHINE/GROUP	Specifies whether the entry is for a computer or a group
Name	String	The NT name of the computer or the group name

Each computer 14 or group of computers 14 within network 12 includes an entry in this table.

2. Links

The Links Table is used to define which computers or groups are contained in other groups. It is defined as follows:

NAME	TYPE	DEFINITION
Parent	String	The group name which contains the computer or group in the Child attribute for this record
Child	String	The group or computer name which is part of the group in the Parent attribute for this record

3. Registry Settings

The Registry Settings Table contains all of the managed keys and values for the entire system. For "WINDOWS NT" implementations, the Registry Settings Table may be the Registry or a portion thereof. The Registry Settings Table is defined as follows:

NAME	TYPE	DEFINITION
Name	String	Computer or group name
Key	String	Registry Key
Allow Override	Yes/No	Indicates which settings may be overridden.
Data	Binary	Buffer which contains data of the type associated with the Name

According to the invention, the "Allow Override" field shown in the above table allows predetermined settings, such as Password Expiration Duration (PED) to be overridden globally. In such instance, the administrator may override all of computers 14 to set the global configuration setting of every computer 14 on the network to have the same value. In the above example, if it is desired to have each password of each computer (and user) on the network set to expire after a predetermined duration, such as 14 days, the above "Allow Override" field would be set to "No" regarding the particular PED setting so that a local override action would not be allowed for this particular setting. Apart from this type of global command wherein a new particular setting (e.g., PED = 14 days) commands the change throughout the entire network without exception, any configuration setting that is "closer" to the computer (see Fig. 8, and description below) takes precedence over any new "non-global" setting.

B. Generation of Already-Have (AH) and Should-Have (SH) Lists

Referring to Fig. 2, the corresponding registry information is available for each machine, such as computers 14, and for each group of computers 14 within network 12 to provide functionality to each of computers 14, for example, to have unique features, common features, and/or mandatory features. For computers 14 labelled A-D, each computer has associated Registry settings specifying, for example, buffer size, domain name, and password

expiration times. The registry information shown in Fig. 2 is generated in real time (i.e., "on-the-fly"), or is periodically stored in an Already-Have (AH) list as a file in configuration database 20, and may be routinely accessed by configuration server 18 to establish and control various features of network 12 and its components; that is, computers 14. For example, all of the computers labelled A-D in Fig. 2 may be required to have a common password expiration time such as 14 days. Otherwise, different attributes may be unique or may indicate groups of computers with common functionality for specific attributes, such as having common domain names for TCP/IP connections, or common buffer sizes.

Referring to the method shown in Fig. 5 and described in greater detail herein, the disclosed configuration management system 10, either automatically or in response to commands and selections from an administrator, reads and/or retrieves the AH lists corresponding to the settings of each of computers 14 shown in Fig. 2 and forms groups of computers therefrom which have common attributes. As shown in Figs. 3, new groups E-H may be generated, such as group E associating all computers having the domain name R2K.COM, and group F associated with computers having the domain name LAB.NET, which may be a test domain name for use in a lab for computer system testing and research. Accordingly, computers A and D are associated with group E, and computers B and C are associated with group F.

As shown in Fig. 3, group G may be generated to include all computers with the password expiration set to 14 days, and in this example, group G includes computers A-D. Such attributes common to all computers may reflect globally-set attributes network-wide, such as a common password expiration duration. Otherwise, computers having different attributes are not associated in group G, and such non-group G computers may be included in group G by appropriate modification of the particular group G attribute of the non-group G computer. Such insertion into group G may be performed individually or *en masse*. Also, member computers of group G may have such settings modified to take an individual computer out of group G.

Similarly, group H may be formed to associate all computers having the buffer size equal to 20, such as computers A-C. As shown in Figs. 2-3, computer D is not included in group H, but another group may be formed constituting all computers having the buffer size equal to 70, of which in this case computer D would be the sole member. As shown in Fig. 3, the grouping of computers is not necessarily hierarchical.

Such grouping of computers may not be limited to single attributes. For example, as shown in Fig. 4 wherein four computers A, B, C, and D, are divided according to common configuration settings. Referring to the figure, group E includes all computers (in this example, groups A and D) having a common domain name; R2K.COM. Similarly, group F includes all computers (computers B and C) which have a LAB.NET domain name. Group G is a "global group" and indirectly includes all computers (A, B, C, and D) by including groups E and F, as shown. Group G includes any configuration settings that are common to all computers (A-D), for example, Password Expiration Duration set to 14 days. Finally, group H includes all computers (in this case A, B, and C) which have a common buffer setting, in this example, Buffer = 20. Hierarchical structures and trees of groupings of computers may thus be generated, and groupings of groups may be performed, as shown in Fig. 4, wherein groups E and F form group G.

Figure 4 illustrates that the program of the present invention works efficiently to implement and manage configuration settings throughout an entire network, in part, because it seeks commonality throughout the network of computers and thereby minimizes the number of controlling paths required to control the particular configuration settings of each computer. In contrast to the present system, an inefficient system would manage each computer of a network from a central point on a direct one-to-one basis wherein a central configuration manager controls each individual computer without regard to commonality between computers.

Referring to Figs. 1-4, the settings of computers A-D and the setting of each of the groups E-H form sublists, sub-sublists, etc. of Should-Have (SH) lists which identify

common resources across multiple computers. By constructing such SH lists from the AH lists of computers using configuration management system 10, configuration server 18 may build additional sublists as additional SH lists from the AH lists and sublists generated therefrom. The SH lists and sublists are then stored in configuration database 20 as a master SH list, which may
5 be at least a portion of the Registry in a "MICROSOFT" "WINDOWS NT" NOS environment. Such generation of AH lists, SH lists and sublists provides for efficient configuration management in configuration server 18 by utilizing predetermined and established Registry settings of each of the individual computers 14 to configure the existing network 12.

10 In operation, configuration management system 10 operates using, for example, a configuration management software program, according to the steps shown in Figs. 5-6. Configuration management program performs at least the following functions, which may be performed in any order:

- (a) generation of the SH lists from the AH lists;
- (b) automatic retrieval of the registry values for each computer listed
15 in configuration database 20 for comparison with configuration management system 10;
- (c) automatic reconfiguration of each computer, as necessary, so that all Registry settings match the Registry settings in configuration database 20; and
- 20 (d) automatic modification of the registry settings in configuration database 20 to match the current settings in each computer 14.

C. Generation of the SH lists

25 As described above with reference to Figs. 1-4, configuration management system 10 generates the SH lists from the AH lists according to the method shown in Fig. 5, in which configuration management system 10 (a) scans the AH lists for common settings among

computers in step 22, such as the AH lists shown in Fig. 2; (b) generates a SH list from the common settings in step 24, to form groups such as shown in Figs. 3-4; (c) examines the constituents of each SH list to create higher levels SH lists, such as group G of Fig. 4 and (d) repeatedly generating SH lists from the constituents groupings in step 28.; (e) determines if any more commonality is found in step 30; repeats steps 26-30 until no more commonality is found; and (f) stores the SH lists in configuration database 20 in step 32.

In operation, some Registry settings and attributes may not be extracted and grouped with the groupable settings, and so may be ungroupable or may be grouped later in response to user intervention such as user selections and preferences. For example, specific computers may share a common geographic location and/or a common geographic designation/setting, yet a set of such computers may be independent and ungroupable with the remaining computers having the same geographic characteristics. Accordingly, the disclosed configuration management system 10 may not group all of the computers despite such common characteristics, but configuration management system 10 may include interactive prompts and subroutines for receiving user intervention in the grouping process at a later time, such as a time chosen by the configuration manager.

In another example, network 12 may support predetermined applications such as graphics editors, and so configuration management system 10 may scan computers 14 to determine that such graphic editors are available in computers 14 at a specific geographic location. However, specific computers may be ungroupable and/or may be prevented from being grouped by configuration management system 10 despite the common features of such computers; that is, the shared graphics editors in a common location. Such prohibitions on grouping specific computers or groups may be administrator-specified and able to be modified by the administrator at a later date.

D. Registry Retrieval and Comparison

Using the SH lists, configuration server 18 operates network 12 to maintain the settings of such SH lists of each of computers 14. During operation of network 12, configuration management system 10 determines what the Registry settings should be from the information in configuration database 20 and then interrogates each computer 14 to retrieve the current values for the settings located in each respective computer 14. Such Registry settings may be modified by a user or an administrator during use of the computer 14 and/or during upgrades or repairs of the computer 14.

Configuration management system 10 operates as a configuration manager by comparing the two sets of Registry settings; that is, from the stored SH lists in configuration database 20 generated as described with reference to Fig. 5, and from the current Registry settings of each computer 14. As differences between the Registry settings of each computer and the Registry settings occur, configuration management system 10 identifies the differences between the AH list and the SH list to reconcile the computer configuration with the configuration management settings of network 12.

As shown in Fig. 6, configuration management system 10 performs the steps of getting a record of the settings from computers 14 and groups table in step 34; checking if TYPE = GROUP in the record in step 36; skipping the record in step 38 if the TYPE = GROUP; and then repeating steps 34-36 until TYPE \neq GROUP in order to process Registry settings of the machines/computers 14. The method then checks in step 40 if the TYPE = MACHINE. If not, the detected settings in the retrieved record may correspond to another group being a subgroup of another group. Accordingly, the record is skipped in step 38 if the TYPE \neq MACHINE; and then steps 34-40 are repeated until TYPE = MACHINE in step 40 in order to process Registry settings of the machines/computers 14. The method then determines the registry settings from the record of the detected machine/computer 14 as described below.

The method gets all entries in the registry settings table where the attribute labelled NAME = NAME of a computer from the Computers and Groups Table in step 42, and

the results are stored in a Temporary Registry Settings Table in step 44. The method then searches to find all Parents in the Links Table for which CHILD = COMPUTER in step 46. The results of this search are then stored in a Temporary Links Table in step 48. The method then searches in step 50 to find all Parents in the Links Table for which CHILD = PARENT
5 from step 46, and the results of the search of step 50 are added to the Temporary Links Table in step 52. The method then checks in step 54 whether more Parents are available, and if so, steps 50-52 are repeated until no more Parents are found.

If no more parents are available, for each unique Parent in the Temporary Links Table, the method searches in step 56 to find all the entries in the Registry Settings Table where
10 NAME = PARENT. For each found entry in the Registry Settings Table, the method checks in step 58 to determine if the KEY and NAME are already present in the Temporary Register Settings Table. If present, the method proceeds to step 61; ANY MORE SETTINGS?. If there are more settings, the method proceeds through step 61a; NEXT STEP, and back to step 56 to retrieve another setting. If no more settings are present at step 61, the method proceeds to 62.

15 At step 58, if no more settings are present, the KEY and NAME entries are added to the Temporary Registry Settings Table in step 60, and then the method proceeds to step 61; ANY MORE SETTINGS, as discussed above. The steps 34-61a may be repeated until all unique Parents have been processed.

20 In step 62, the method then accesses the Registry of the specific computer whose NAME was found in steps 34-40. For each KEY and NAME in the temporary registry settings table, the method retrieves the current data in step 62 from the current registry of the specific computer, and the method then compares such current data in step 64 with the DATA from the temporary registry settings table. If it is different, the method displays the difference in step 66; for example, by outputting a line or message, such as a dialog box in a graphic user interface,
25 to the user or administrator of network 12. The message may include the computer name, the registry key and name, the data from the database, and the current data from the machine.

Steps 34-66 may then be repeated for each computer in the computers and groups table. At the end of this method, the user has a complete list of differences between the registry settings in the database and the current settings for each computer 14 listed in configuration database 20.

5 According to another embodiment of the invention, referring to Figs. 6 and 7, an "allow override" setting is included in the method wherein each setting in the temporary registry settings table includes an allow override setting. If the allow override setting is set to "No", then the current setting is replaced by the new setting. If the allow override setting is not set to "No", then the settings are not changed, i.e., the "closer" setting takes precedence as
10 described above. Referring to Fig. 7, steps 56-62 include such steps 59 and 63 to perform the allow override checking. In step 59, the allow override setting is checked to determine if it is set to "No". If it is set to "No", the method proceeds to a REPLACE step 63, wherein the current setting is replaced with the selected setting. After step 63 in Fig. 7, the method proceeds to step 61 of Fig. 6 and continues as shown in Fig. 6 and described above. If, in step
15 59 (referring to Fig. 7), the allow override setting is not set to "No", then the process returns to step 56 of Fig. 6, and proceeds as shown in Fig. 6 and described above.

E. Reconfiguring Computers to Conform to the Configuration Database

20 Once the differences between configuration management system 10 and each computer's current registry settings have been determined, the user at the administrator's computer site may want configuration management system 10 to reconcile those differences by changing the current registry settings on the specific computer to match the settings stored in configuration database 20. The method for reconfiguring each computer includes the step of accessing the specific computer, according to the Computer Name, which corresponds to each
25 difference found during the Registry Retrieval and Comparison operation described above, which includes the Computer Name, Registry Key and Name, Database Data, and Current Data.

In accessing the specific computer given by the Computer Name, configuration management system 10 sets the registry of the computer such that the Registry Key and Registry Name are set to the Current Data for the computer. These steps of accessing each computer and setting its registry are performed for all differences found between the stored Registry settings of network 12 and any specific computer 14.

Configuration management system 10 may be interfaced with an external Version Control System (VCS) such that Registry settings are exported out of configuration database 20 into a format which can be accepted and stored by the VCS. Retrievals from the VCS can be imported into configuration database 20 so that the modifications of configuration database 20 are performed by configuration management system 10 by comparing the database settings with the settings of the current computers 14 for automatically adjusting configuration database 20 or the configuration of the computers.

F. User-Controlled Reconfiguration of Computers

In an alternative embodiment, the differences between the Registry in configuration database 20 and the Registry settings or AH list of each computer are displayed to the user of the corresponding computer to perform user intervention and reconciliation of the differences. Accordingly, in step 66 above, the differences are displayed to a GUI associated with the user such as a GUI on a display of the user's computer. At that time, the user may then determine if the current Registry settings of the specific computer should be conformed to or reconciled with new Registry settings specified by the Registry in configuration database 20.

Such differences may be indicated by displaying to the user the AH list of the user's computer and the SH list specified by the Registry in configuration database 20. Upon approval or other commands from the user, configuration management system 10 may automatically perform the reconciliation by, for example, updating the registry of the computer. One option available to the user is the ability to command configuration management system 10

to automatically accept the AH of the user's computer into configuration database 20. Alternatively, for a SH list conflicting with the AH list of the user's computer, the user may command configuration management system 10 to delete the entry of the user's computer from the conflicting SH list, thus removing the conflict. Both reconfiguration options selectable by the user may require configuration management system 10 to revise the SH lists of network 12. Such optional and selectable reconfiguration methods improve the management of network 12, since the SH lists may include errors causing SH discrepancies with the user's computer, which the user may remedy as such SH discrepancies occur.

Some computers may have special assignments so that their configuration must be different, as is understood by the user at the local level, or perhaps at the domain level. In such instances, the particular settings of the user's computer would not be conformed and the current Registry settings of a specific computer would remain.

In an alternative embodiment, an administrator may annotate the entries in the Registry in configuration database 20 for any of computers 14, groups of computers, or Registry settings to record a reason why any changes were made. These annotations may be stored as separate entries in configuration database 20 which automatically store the date and time, the administrator's system identification, as well as the annotations made, which may be in the form of a text message. Configuration management system 10 may also generate reports on the history of changes to configuration database 20 according to any overriding user selections and/or according to the stored annotations.

G. Precedence-Controlled Reconfiguration of Computers

The SH lists may correspond to groups, and so SH lists of computers of one group may be subsets of a SH list of an encompassing group. As groups are modified and computers 14 are incorporated or removed from network 12, and even as Registry settings are modified, SH lists throughout network 12 may be generated listing different settings for a

particular computer. Accordingly, such conflicting SH lists may adversely affect the performance of network 12.

Referring to Fig. 8, an illustrative embodiment is shown to define term "closer" and "closeness" used above. In this example, computers A, B, C, and D are shown grouped according to common configuration settings, as discussed above. Computers A and D are grouped together, in this case, because both of their password expiration settings are set to PE=7 days. Similarly, computers B and C are grouped together because they have other common configuration setting (not shown). Groups E and F are grouped together as sub-group G because, in this example, the configuration manager wants to change the password expiration setting of all of the computers A-D to PE=14 days. Computers A and D will have their password expiration setting changed to PE=14 days, automatically, but only if the allow override is set to "no". If the allow override is set to "yes", group E, which effectively has a password expiration setting of PE=7, will override the setting change request from the configuration manager (group G) and computers A and D will retain their PE=7 setting. This is because group E is "closer" in hierarchy to the computers A and D, and if allowed, it is preferred that the hierarchal level which is "closer" to a computer take precedence in configuration setting changes.

Each local site "educates" the enterprise level regarding Registry settings as they may be "tested" in the field, and so the enterprise level controlled by configuration management system 10 respects the selected operability of the locally determined Registry settings. Accordingly, priority and cardinality values may be adjusted to control the precedence of SH lists.

An important feature of the disclosed configuration management system 10 and method is that the registry files of a large number of computers, for example, thousands of interconnected computers, may be set to desired and customized configurations so the individual computers efficiently work together in a distributed computing environment. If particular files

at any local or domain site throughout an enterprise become corrupt, and Registry settings become faulty, implementation of the disclosed configuration management system 10 and method manages and effectively "resets" the configuration files of each computer so that any local problems can be corrected.

- 5 While the invention has been particularly shown and described with reference to a preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention.

WHAT IS CLAIMED IS:

1. A method for managing the configuration settings of a plurality of computers in a network, the method comprising the steps of:

processing the configuration settings of each of the computers as Already-Have (AH) lists to extract common configuration setting groups;

generating Should-Have (SH) lists from such common configuration setting groups;

storing the SH lists in a configuration database as a master configuration SH list; and

managing the configuration settings of the plurality of computers during operation of the network using the master configuration SH list.

2. The method of claim 1, wherein the step of managing includes the steps of: scanning the current configurations settings of each of the computers;

comparing the current configuration settings with the master configuration SH list to detect differences; and

resolving the differences between the current configuration settings and the master configuration SH list.

3. The method of claim 2, wherein the step of resolving includes the step of: prompting a user to select a resolved set of configuration settings from the current configuration settings and the master configuration SH list; and

automatically adjusting one of the current configuration settings and the master configuration SH list in response to the selection of the user.

4. The method of claim 2, wherein the master configuration SH list includes a first SH list and a second SH list associated with the first SH list;

wherein the step of scanning includes the step of:

5 detecting for the current configuration setting of a first computer as having the settings of the second SH list;

wherein the step of comparing includes the step of:

detecting for differences between the first and second SH lists; and

wherein the step of resolving includes the step of:

10 automatically adjusting the master configuration SH list to include the settings of the second SH list.

5. The method of claim 2, wherein the master configuration SH list includes a plurality of SH lists having associated settings;

wherein the step of comparing includes the steps of:

15 comparing the current configuration settings to at least one of the SH lists,
and

determining the current configuration settings of a first computer to be closest to the configuration settings of a first SH list; and

wherein the step of resolving includes the step of:

20 automatically adjusting the master configuration SH list to specify the configuration settings of the first computer to be the configuration settings of the first SH list.

6. A system for managing the configuration settings of a plurality of computers in a network comprising:

means for processing the configuration settings of each of the computers as Already-Have (AH) lists to extract common configuration setting groups;

5 means for generating Should-Have (SH) lists from such common configuration setting groups;

means for storing the SH lists in a configuration database as a master configuration SH list; and

10 means for managing the configuration settings of the plurality of computers during operation of the network using the master configuration SH list.

7. The system of claim 6, wherein the managing means includes:

means for scanning the current configurations settings of each of the computers;

15 means for comparing the current configuration settings with the master configuration SH list to detect differences; and

means for resolving the differences between the current configuration settings and the master configuration SH list.

8. The system of claim 7, wherein the resolving means includes:

20 means for prompting a user on an associated display of one of the computers to select a resolved set of configuration settings from the current configuration settings and the master configuration SH list; and

means, responsive to the selection of the user, for automatically adjusting one of the current configuration settings and the master configuration SH list.

9. The system of claim 7, wherein the master configuration SH list includes a first SH list and a second SH list associated with the first SH list;

wherein the scanning means detects for the current configuration setting of a first computer as having the settings of the second SH list;

5 wherein the comparing means detects for differences between the first and second SH lists; and

wherein the resolving means automatically adjusts the master configuration SH list to include the settings of the second SH list.

10 10. The system of claim 7, wherein the master configuration SH list includes a plurality of SH lists having associated settings;

wherein the comparing means compares the current configuration settings to at least one of the SH lists, and determines the current configuration settings of a first computer to be closest to the configuration settings of a first SH list; and

15 wherein the resolving means automatically adjusts the master configuration SH list to specify the configuration settings of the first computer to be the configuration settings of the first SH list.

11. A network comprising:

20 a plurality of computers;
a configuration server for controlling the configuration settings of the plurality of computers;

a configuration database for storing the configuration settings of the plurality of computers in Should-Have (SH) lists as a master configuration SH list;

25 a configuration management system, operating with the configuration server, for managing the configuration settings of the plurality of computers in a network, for processing

the configuration settings of each of the computers as Already-Have (AH) lists to extract common configuration setting groups, for generating the SH lists from such common configuration setting groups, and for managing the configuration settings of the plurality of computers using the master configuration SH list.

5

12. The network of claim 11, wherein the configuration management system includes:
means for scanning the current configuration settings of each of the computers;
means for comparing the current configuration settings with the master
configuration SH list stored in the configuration database to detect differences; and
10 means for resolving the differences between the current configuration settings and
the master configuration SH list.

10

13. The network of claim 12, wherein the resolving means includes:
means for prompting a user on an associated display of one of the computers to
15 select a resolved set of configuration settings from the current configuration settings and the
master configuration SH list; and
means, responsive to the selection of the user, for automatically adjusting one of
the current configuration settings and the master configuration SH list.

15

14. The network of claim 12, wherein the master configuration SH list includes a first
SH list and a second SH list associated with the first SH list;
wherein the configuration management system detects for the current configuration
setting of a first computer as having the settings of the second SH list, and detects for
differences between the first and second SH lists, and automatically adjusts the master
25 configuration SH list to include the settings of the second SH list.

20

25

15. The network of claim 12, wherein the master configuration SH list includes a plurality of SH lists having associated settings;

wherein the comparing means compares the current configuration settings to at least one of the SH lists, and determines the current configuration settings of a first computer to be closest to the configuration settings of a first SH list; and

wherein the resolving means automatically adjusts the master configuration SH list to specify the configuration settings of the first computer to be the configuration settings of the first SH list.

16. The network of claim 11, wherein the configuration management system is a software program executed by the configuration server.

17. The network of claim 11, wherein the configuration server executes a "WINDOWS NT"-based network operating system for operating the plurality of computers.

FIG. 1

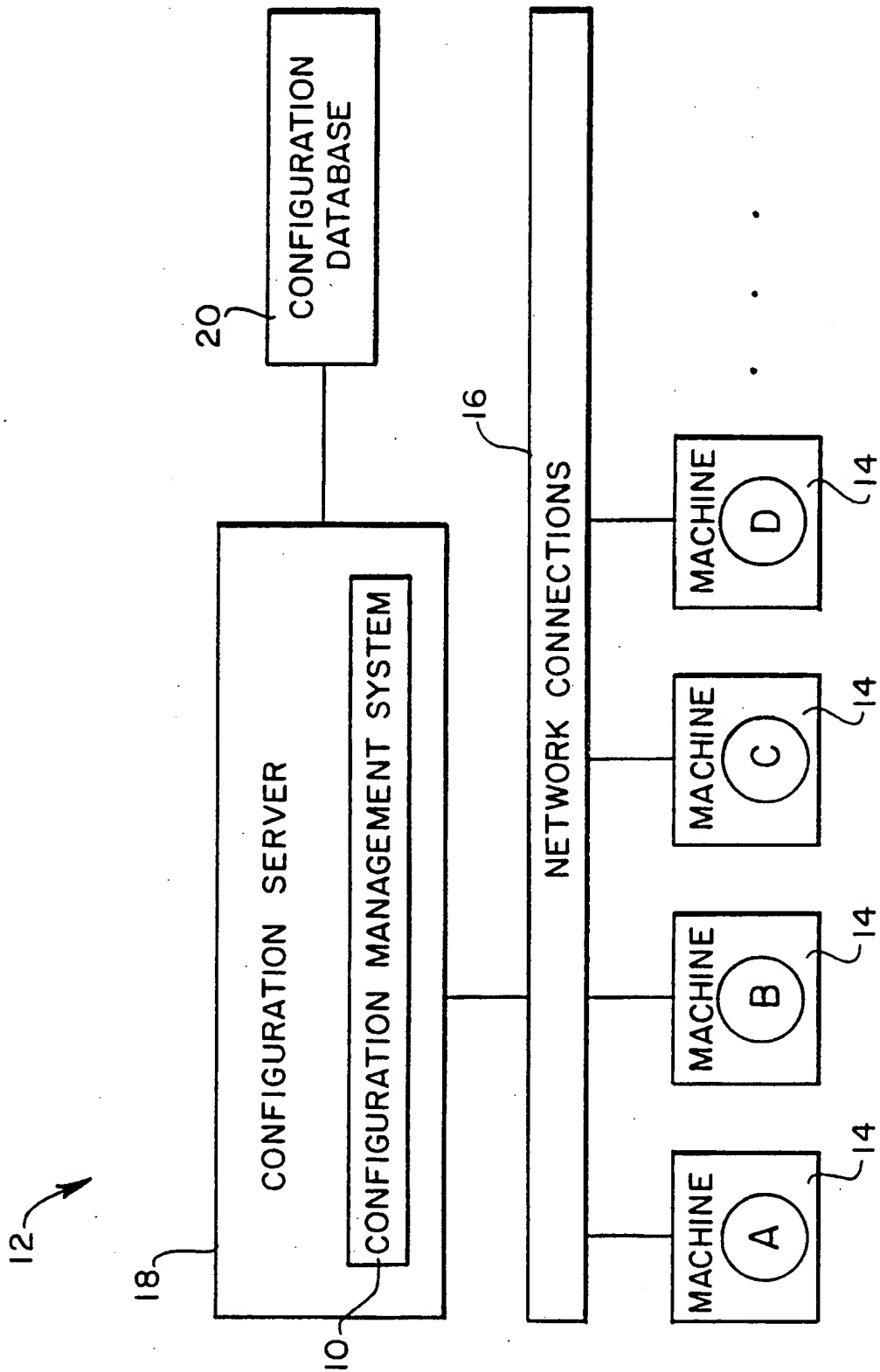


FIG. 2A

A

Buffer = 20
Domain Name = R2K.COM
Password Expiration = 14

FIG. 2B

B

Buffer = 20
Domain Name = LAB.NET
Password Expiration = 14

FIG. 2C

C

Buffer = 20
Domain Name = LAB.NET
Password Expiration = 14

FIG. 2D

D

Buffer = 70
Domain Name = R2K.COM
Password Expiration = 14

FIG. 3

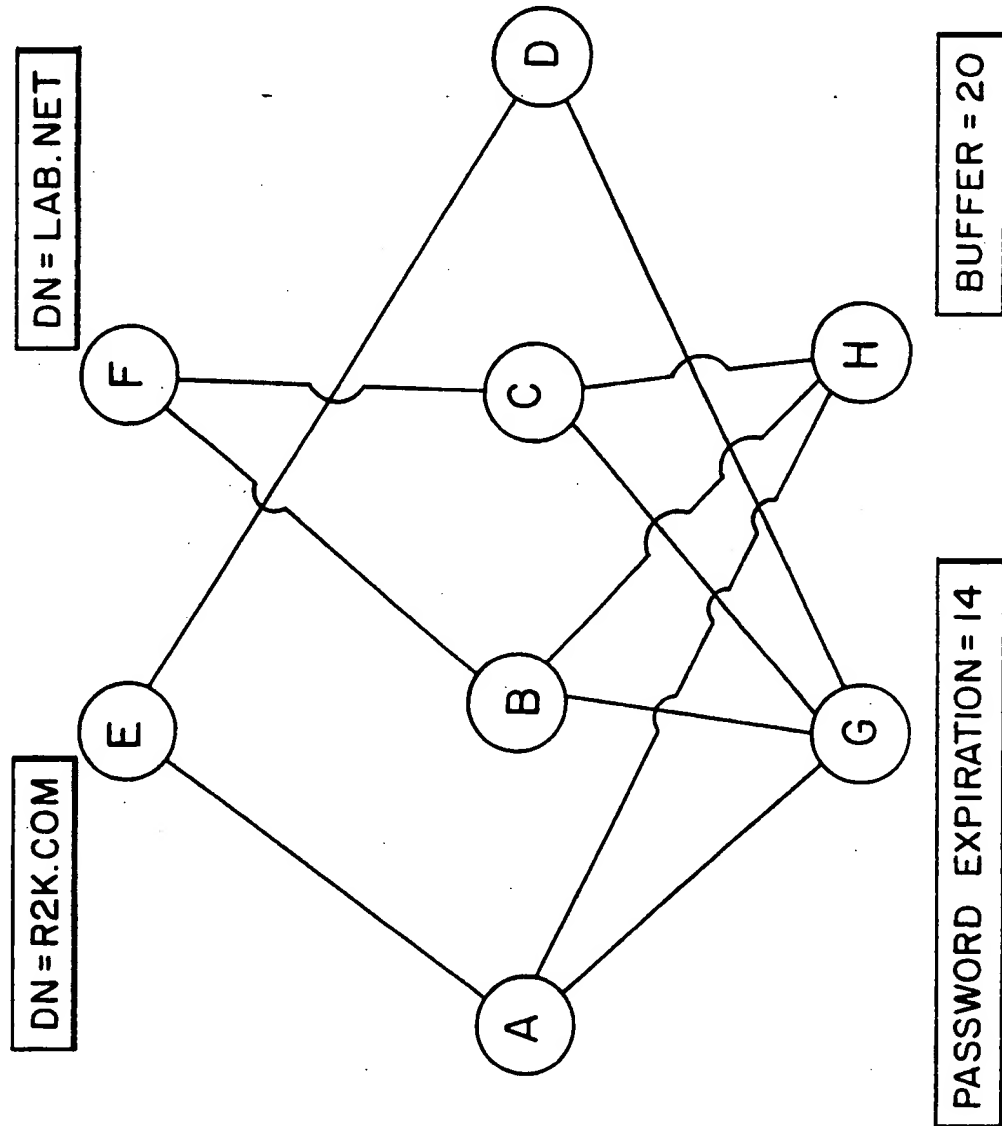
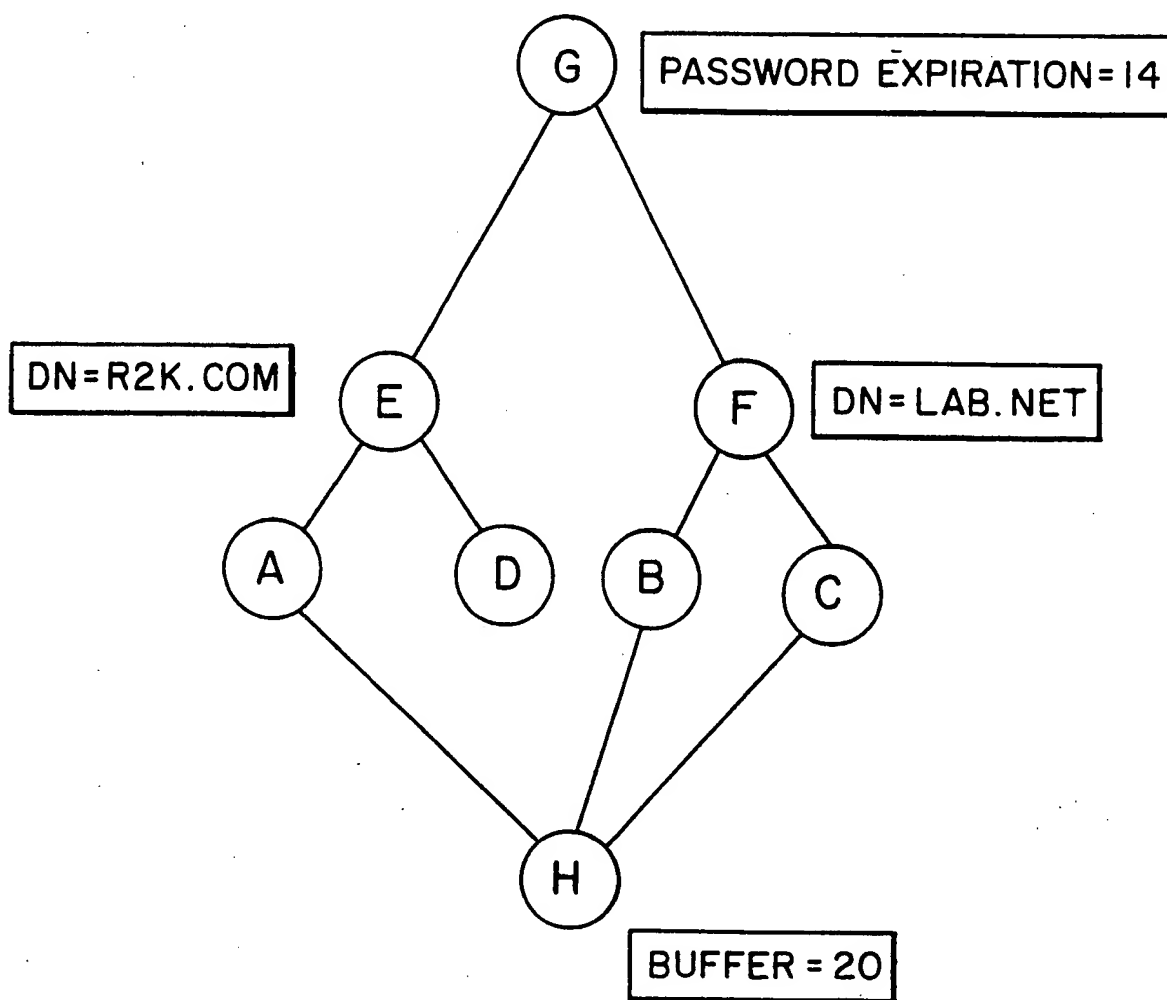
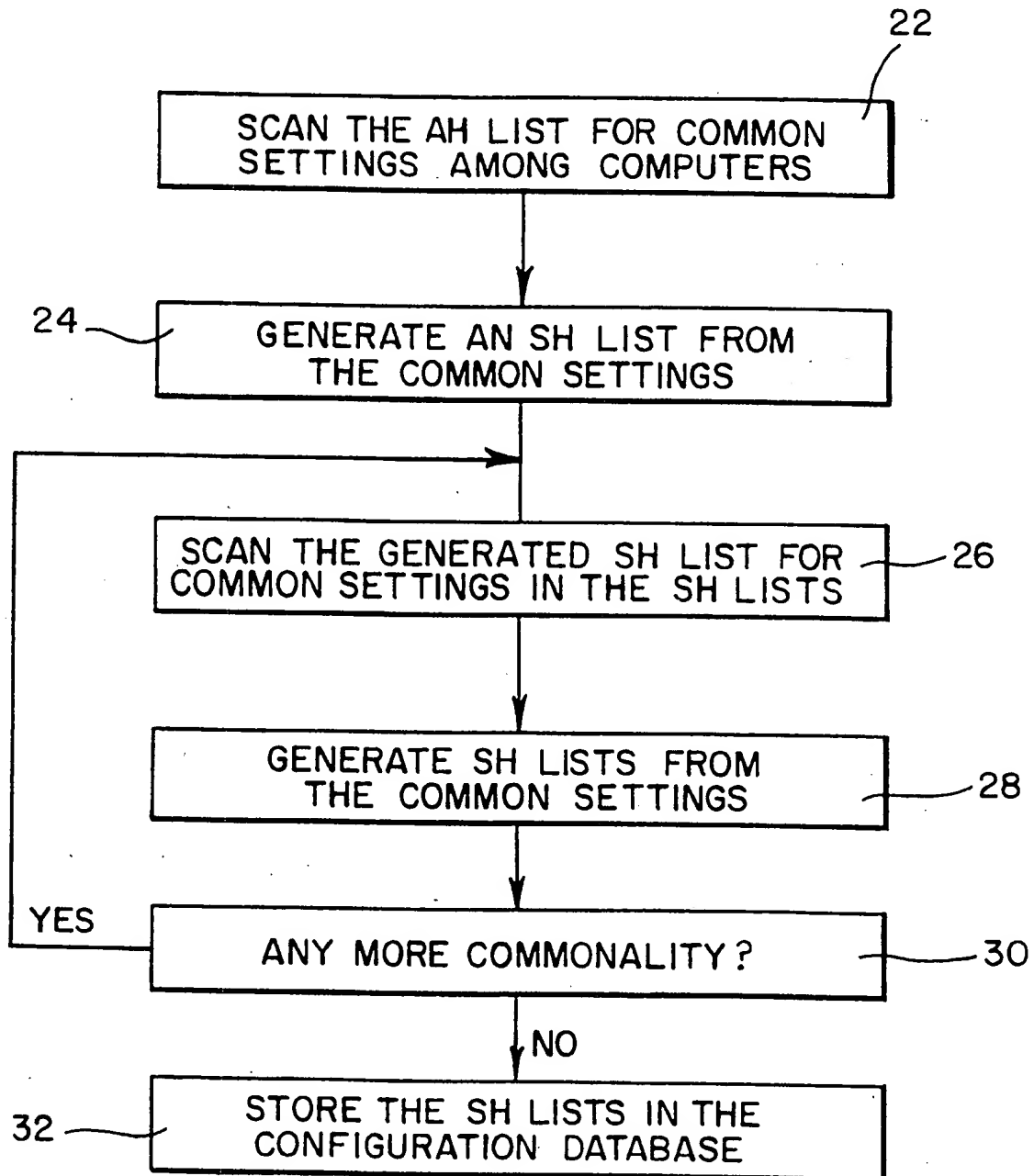


FIG. 4



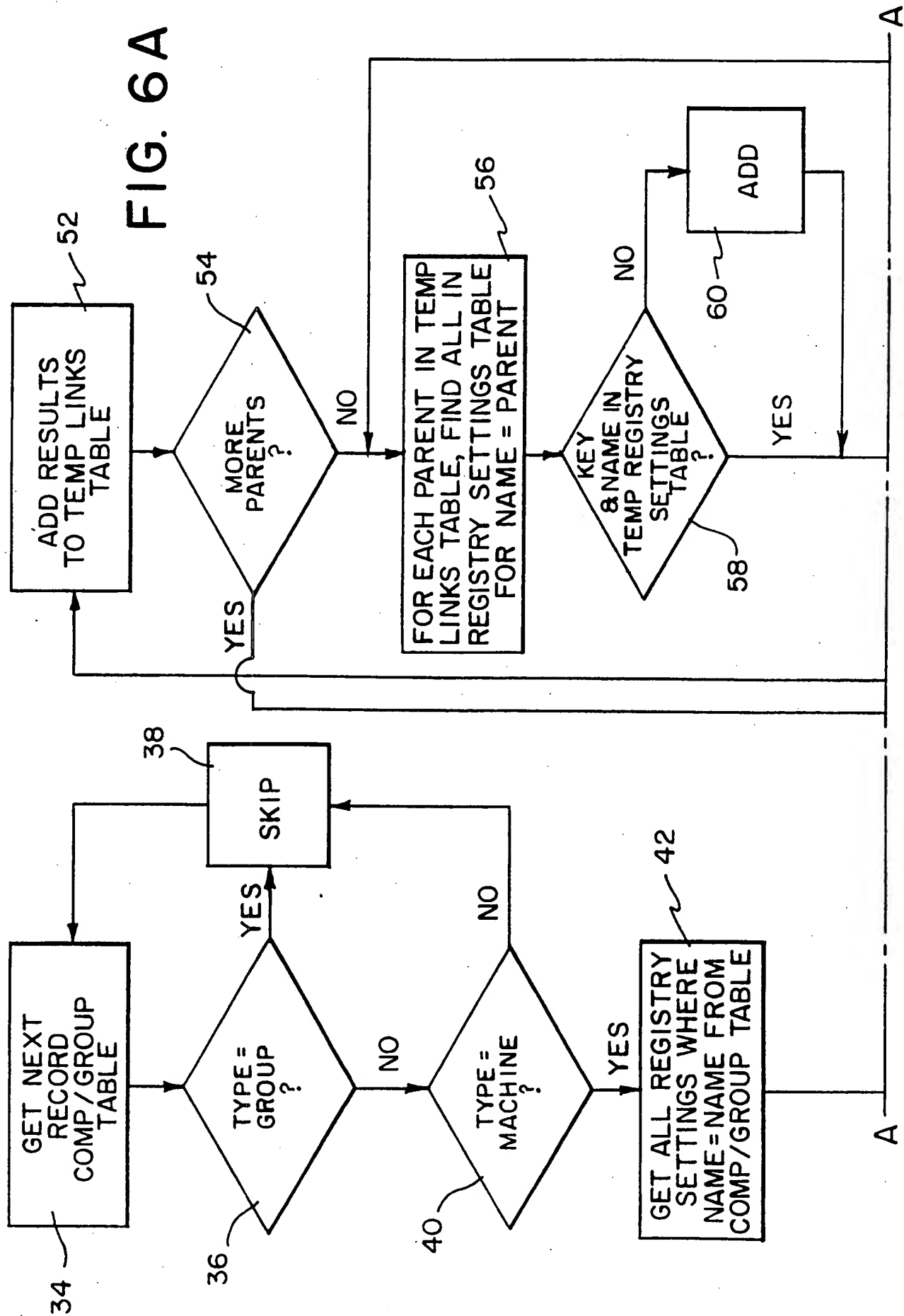
5/9

FIG. 5

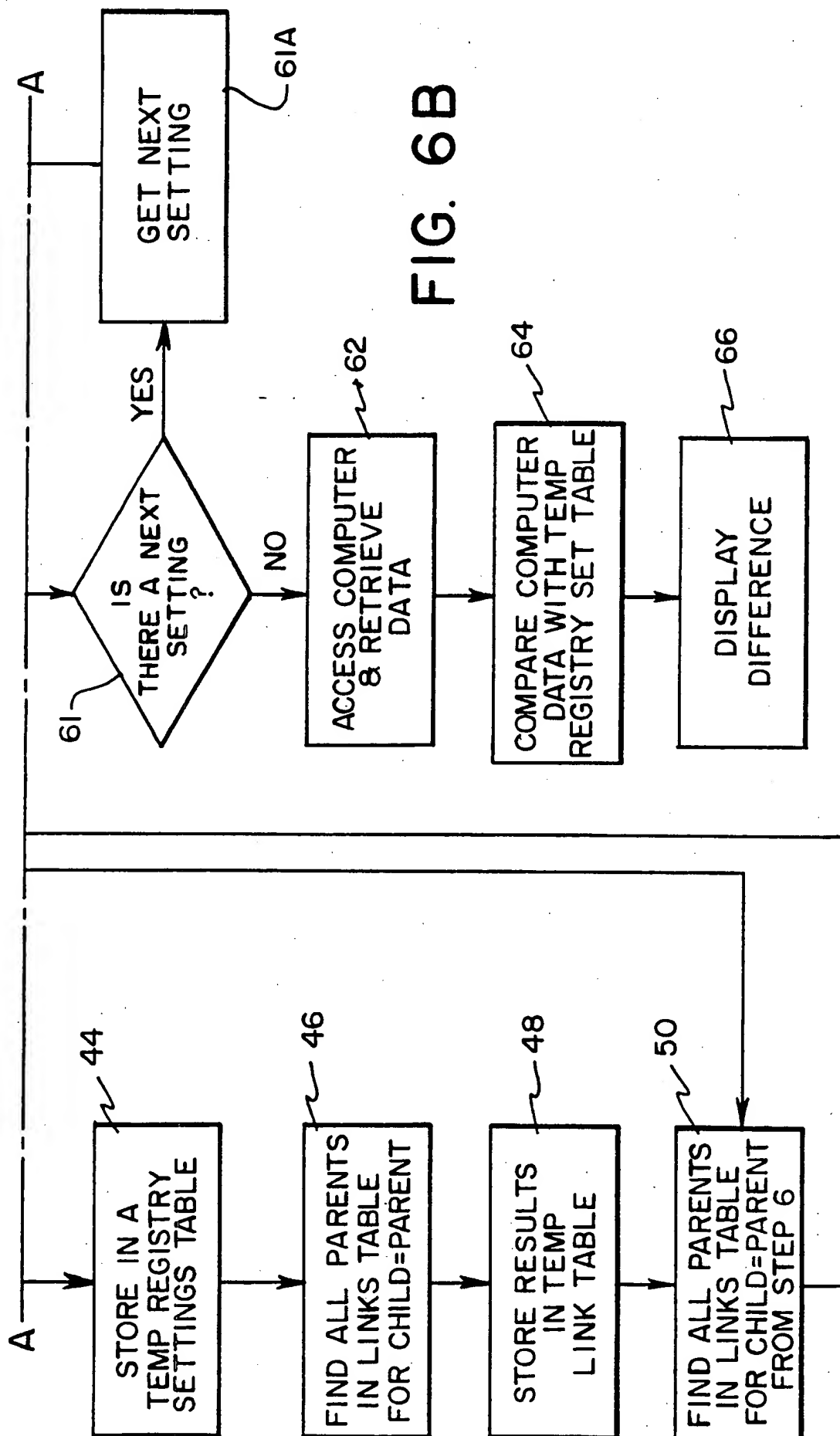


6/9

FIG. 6A

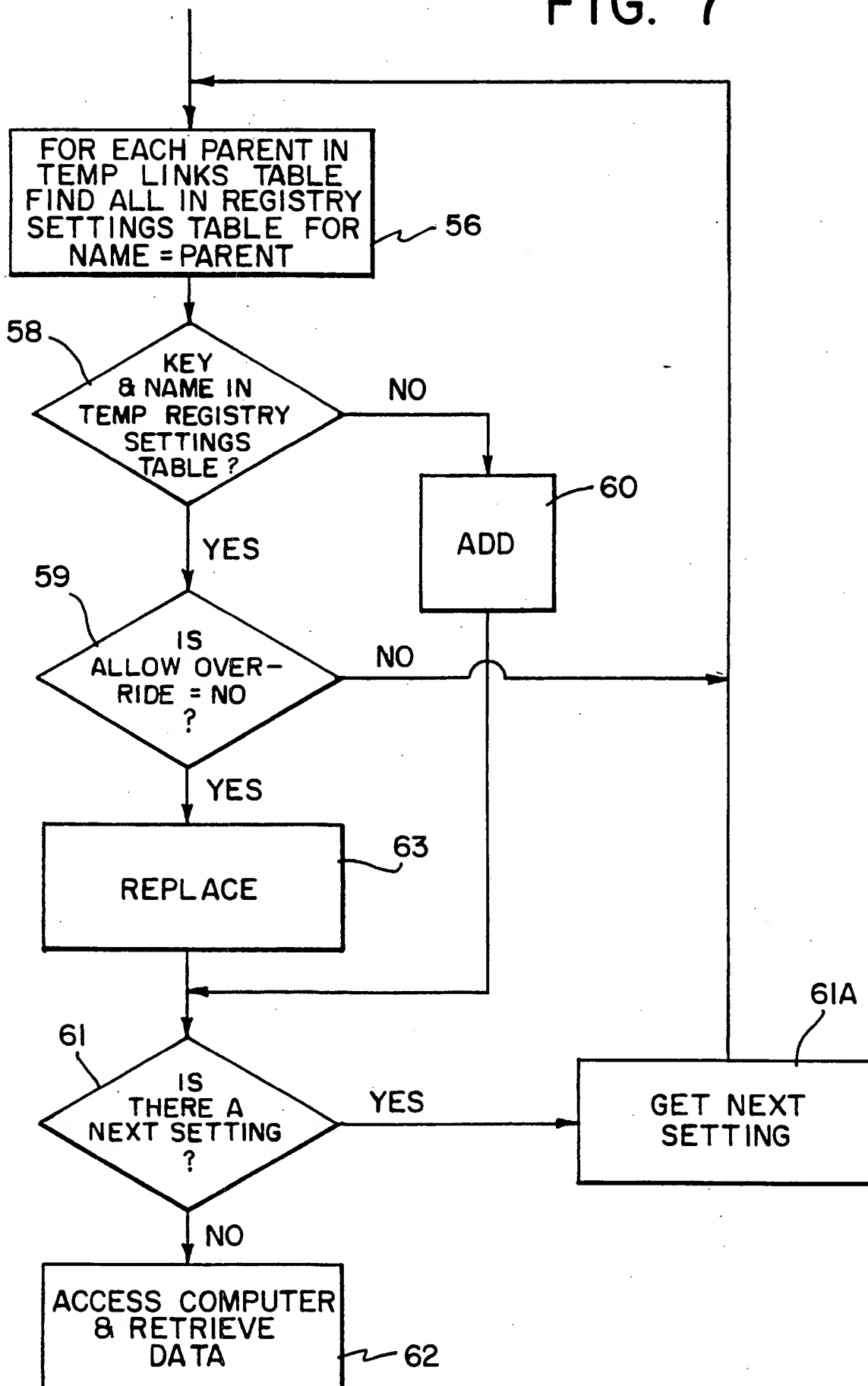


7/9



8/9

FIG. 7



9/9

FIG. 8

